Z-Wave 보안 취약점 분석

임지환[1] · 조기윤[1] · 성민석[1] · 정영호[1] · 김성범[1] · 김경곤[1] · 이경문[1] · 서화정[2]

[1]한국정보기술연구원 차세대 보안리더 양성 프로그램

[2]한성대학교 IT융합공학부

Ji-Hwan Lim[1] · Ki-Yoon Cho[1] · Min-Suk Sung[1] · Sung-Bum Kim[1] · Young-Ho Jung[1] · Anesra[1] · GilGil[1] · Hwa-Jeong Seo[2]

[1]Best of the Best

[2]Department of IT, Hansung University, Seoul, 02876 Korea

요 약

현재 IoT 기기들은 각각의 기능과 필요에 맞는 무선 통신을 위하여 Wifi, BLE, Z-Wave, Zigbee, NFC 등의 여러 프로토콜을 사용한다. 본 논문은 그중 최근 홈 IoT 시장에서 주목받고 있는 사용성과 확장성이 좋고 저전력 프로토콜인 Z-Wave 무선 통신 프로토콜을 사용하는 제품들의 보안 취약점을 기술한다. 본 논문은 Z-Wave에 대한 관련 연구 및 동향을 소개하며 직접 분석한 국내 제품들에 대한 보안관점의 문제점을 간략히 나타내고, 몇 가지 공격 형태에 대해 설명한다. 또한, Z-Wave 프로토콜에 대해 분석하여 어떤 데이터를 주고받는지 기술한다. 본 논문을 통해 무선 사물인터넷 시장에 대한 보안성을 높일 수 있는 방안을 연구하는 좋은 출발점이 되었으면 한다.

I. 서론

Z-Wave는 통신, 네트워크 및 응용 계층 프로토콜을 포함하는 IoT 최적화 기반의 구현이다. 사용자는 Z-Wave 센서, Actuator, 컨트롤러, 라우터 및 인터넷 게이트웨이를 구성하여 스마트 홈 및 사무 자동화 시스템을 구축할 수 있다. 스마트 홈 자동화 시스템은 난방, 환기 및 에어컨, 조명 및 물리적 보안 시스템을 위한 중앙 집중식 제어 및 모니터링 기능을 제공한다. 이러한 중앙제어 시스템, 보안 센서 및 경보 시스템과 같은 다양한 가정 장치는 무선 또는 유선 통신 링크를 통해 한 개의 컨트롤러 장치에 최대 232개의 노드가 있는 Mesh 네트워크를 형성하고 이를 통해 스마트 홈을 형성한다.

Z-Wave의 프로토콜 계층은 물리, 전송, 라우팅, 어플리케이션 크게 4가지 계층으로 구성되고, 통신 기반은 물리, 맥 계층이 ITU.G.9959에 정의되어 있는 독점 무선 스택을 사용한다. 또한, 애플리케이션 계층의 상당 부분이 OpenZwave [1]를 통하여 오픈 소스코드로 공개 되어있다.

스마트 홈 및 사무공정 자동화 시스템을 위해 최적화되어 효과적으로 운용되는 시스템에도 불구하고, 타 프로토콜과 마찬가지로 공격자가 무선 패킷을 쉽게 Sniffing 하거나, Spoofing 할 수 있다는 단점이 있고, 이러한 시스템을 지속시키기 위해서는 무선 통신의 기밀성과 무결성을 보호하는 것이 매우 중요하다. 따라서 키 설정, 암호화 방식, 프레임 무결성 보호 및 장치 인증과 같은 보안 서비스가 사용되고 있다. 따라서 본 논문에서 Z-Wave 사용 보안성 강화를 위해 국내에서 사용하는 제품의 취약점을 분석한다.

본 논문의 구성은 다음과 같다. 2장 관련 연구 동향에서 Z-Wave 프로토콜에 대해 분석하고, 3장 본론에서 Z-Wave 사용 취약점에 대해 분석한다. 그리고 4장 성능평가에서 DFD를 기

반으로 몇 가지 공격벡터를 기술하고, 마지막 5장에서 본 논문의 결론을 맺는다.

II. 관련 연구 동향

2.1 Z-Wave 프로토콜 분석

모든 Z-Wave PAN (Personal Area Network)에는 Z-Wave 컨트롤러에 내장되어있는 고유의 Home-ID가 있다. 이 Home-ID는 제조 시에 설정되고 유저에 의해서 변경될 수 없도록 설계되어있다. 이렇게 하는 이유는 특정 Home-ID를 갖는 컨트롤러가 인접 Z-Wave 네트워크를 침범하지 않기 위해서 이다. 따라서 Z-Wave PAN 내부의 모든 노드는 컨트롤러와 동일한 Home-ID와 Node-ID가 조합되어 고유하게 식별된다. 만약 서로의 범위 내에 여러 Z-Wave 컨트롤러가 작동한다면 각각의 장치는 자신이 속한 컨트롤러의 Home-ID에 의해 식별될 수 있을 것이다. Z-Wave의 기본 프레임 구조는 그림 1과 같다. Z-Wave MAC 프레임의 기본 구조는 아래와 같이 P, SOF, User-Data, EOF로 구성된다. 또한, Broadcast, Multicast, 라우팅과 탐색기 프레임은 User-Data 섹션 내부의 Home-ID, Src, Dst 등 다른 필드를 사용한다. Application Data는 실질적으로 사용자가 전달하고자 하는 Z-Wave Comand Class 명령 또는 메시지 전달에 사용된다. Application Data는 AES-128로 암호화 되어 통신한다 [2].

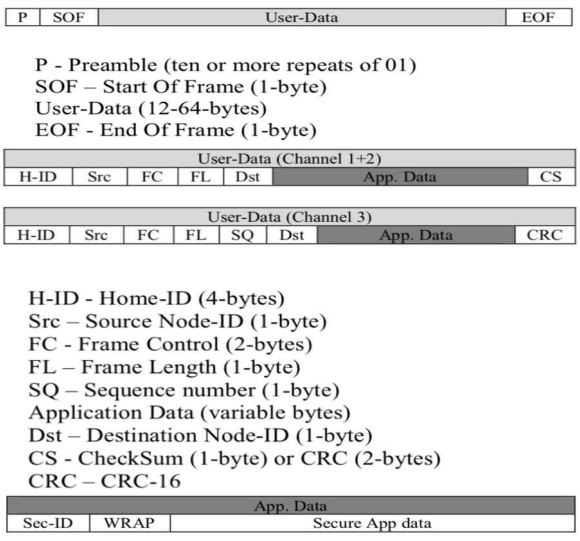

Fig. 1 Z-Wave packet frame

2.2 Z-Wave 보안 프로토콜 등급

Z-Wave의 초기의 100 시리즈는 TDES (Triple Data Encryption Standard)가 있었고, 암호화는 항상 Vendors에게 맡겼다. 모든 Vendors가 암호화 옵션을 사용하지는 않았기 때문에, Z-Wave 장치의 취약점을 보여주는 연구가 지속적으로 있어왔다. 따라서 Z-Wave는 S2 보안 프레임워크로 인증을 대체할 것을 알렸다. S2 보안 프레임워크는 데이터링크용 AES-128과 키 교환을 위해 ECDH를 기반으로 한다. 간단히 정의하면 Z-Wave S2 보안 프레임워크는 기존의 S0 방식의 키 교환 부분의 취약한 부분을 ECDH 암호화를 적용하여 문제를 해결한 것으로 볼 수 있다. Z-Wave S0 키 교환 및 통신 과정은 그림 2와 같고, 사용되는 암호화 방식은 표 1과 같다.

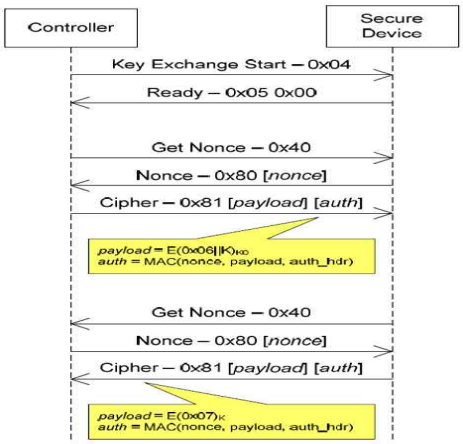

Fig. 2 Key exchange protocol (S0) [3]

Table. 1 Encryption method used in Z-Wave S0 security

$K_c = AES-ECB_{k_n}(Passwd_c)$
$K_m = AES-ECB_{k_n}(Passwd_m)$
$C = AES-OFB_{k_c}(IV, P)$
$MAC = AES-CBCMAC_{k_m}(IV, SH \| SRC \| DST \| LEN \| C)$

III. 본론

본 장에서는 국내 Z-Wave 제품의 보안성 검사와 Z-Wave 보안 취약성 부분을 기술한다. 우선 테스트를 위한 장비는 Silicon Labs에서 판매하는 Dev Kit [4]을 구매하였다. Z-Wave

무선 패킷을 분석하기 위한 EU (868MHz), US (908MHz), Asia (920MHz) 대역의 패킷을 Sniffing할 수 있는 동글 (Zniffer)은 Dev Kit에서만 구할 수 있다. Zniffer를 통하여 작동하고 있는 무선 패킷을 Wireshark와 유사한 Slicon Labs에서 제공해주는 App을 통해서 분석할 수 있다.

3.1 국내 Z-Wave 제품 보안성 검사

국내 Z-Wave 제품의 취약점 분석 및 테스트를 수행하기 위해 망 구성도를 그림 3과 같이 구성하였다.

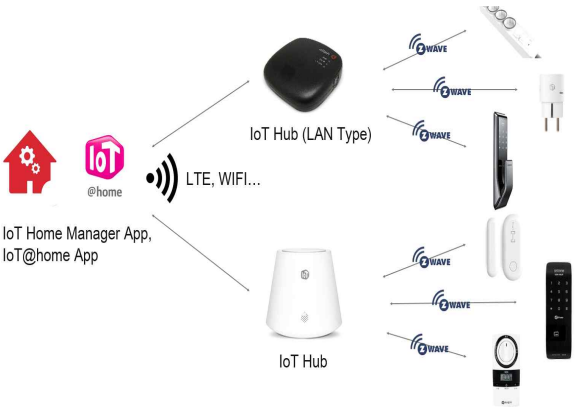

Fig. 3 Overall network configuration for analyzed products

분석 결과 국내 Z-Wave 제품들은 대부분 Non Secure 모드 이거나, S0 보안 방식을 채택하여 취약한 상태임을 확인하였다. 2016년 7월 Sigma Designs에서 Z-Wave S2 보안 방식을 공표 [5]하였지만, 아직까지는 국내에서는 S2 보안 방식을 사용하는 컨트롤러는 구하기 힘든 상황이다.

공격 시연 테스트는 현재까지는 2가지 방식으로 진행하였다. Sigma Designs에서 제공해주는 Z-Wave PC Controller 5 App을 변조하여 Spoofing 툴로 사용하였다. 우선 공격을 수행하기 위해서 필요한 정보인 Home-ID를 Sniffing을 통해 탈취한 후, 해커가 자신의 Z-Wave 컨트롤러의 Home-ID를 Target Home-ID로 변조하고, 나머지 패킷을 수정하여 명령을 전송하였다. 그 결과 성공적으로 공격이 수행되는 것을 확인하였다.

사용자가 보내는 Z-Wave 패킷은 그림 1의 User-Data 부분이다. 해당 공격 방식이 의미 있는 이유는 Asia 대역에 아직까지는 공격 툴이 없기 때문이다. 기존의 공격은 Z-Force Attack을 통하여 수행할 수 있었지만, 해당 공격은 EU, US 대역에서만 사용할 수 있다. 따라서 국내 대역에서 Z-Wave 공격을 성공하기 위해서 오픈 소스를 참조하여 Non-Secure, S0, S2까지 공격할 수 있는 새로운 GUI Type Spoofing Tool을 구현하였다. HackRF를 사용하는 공격 방식으로는 Replay Attack을 시연하였다. 해당 방법은 현재까지는 Z-Wave Non-Secure Mode에서만 적용한 상태이다.

3.2 Z-Wave 보안 취약성

국내 Z-Wave를 사용하는 Node 제품에 임베디드 장비 취약점을 활용한 공격으로 그림 4와 같은 방식으로 Flash Memory에 있는 펌웨어를 추출해본 결과 일부는 Flash Memory에 접근할 수 있는 Uart, SPI 통신 디버깅 포인트를 사전에 끊어놓은 상태로 제품을 출시하여 접근할 수 없었다. 하지만, 일부는 접근이 허용되는 부분이 발견되어 Z-Wave 보안등급 S0 기준으로 초기에 설정되어있는 $Passwd_c$, $Passwd_m$를 탈취하였다. S0 방식은 공개키 암호화 방식이 아닌 대칭키 암호화 방식이므로 공격자가 페어링 시 Sniffing으로 정보를 탈취하거나, 강제 재 페어링 공격에 성공한다면 PAN에서 모든 노드를 공격자가 원하는 대로 제어할 수 있는 취약점이 있다.

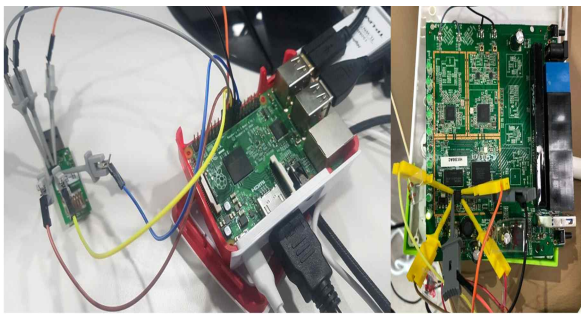

Fig. 4 Exploitation of firmware extraction on embedded equipment

DFD는 분석 대상 시스템의 데이터 흐름을 추상적으로 보여주는 방법으로 Z-Wave S0 보안 프로토콜에 대해 그림 5와 같이 간략히 나타내었다. 이에 대해 설명하면 과정 1은 사용자

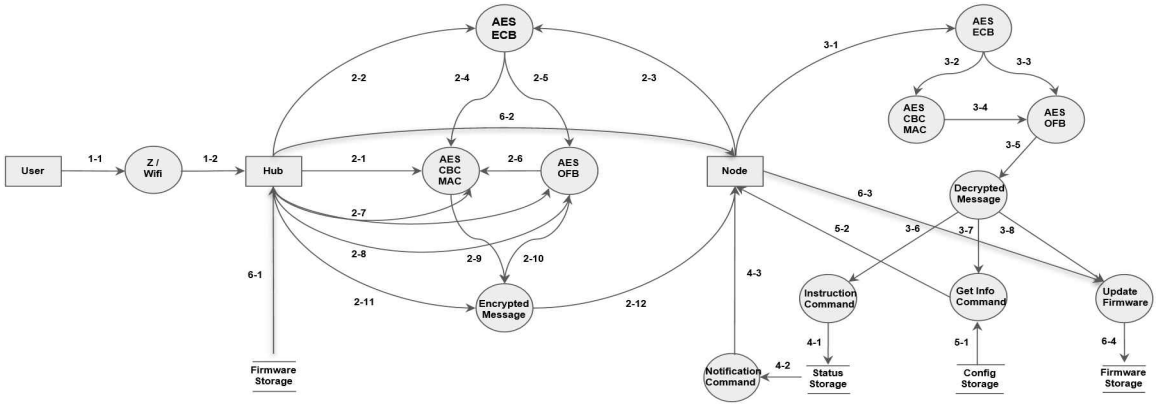

Fig. 5 Z-Wave S0 Data Flow Diagram (DFD)

와 Hub 간의 통신이고, 과정 2, 3은 허브와 노드간의 통신 데이터 송수신 과정에 사용되는 암호화 방식이다. 과정 4, 5, 6은 Controller에서 받은 명령을 노드에서 처리하는 방식이다. DFD 도식화 과정을 거치고 난 뒤, 취약점을 식별해 보니 명령 데이터를 보내는 부분에서 취약점이 주로 발생하는 것으로 확인하였다.

IV. 성능평가

본 장에서는 식별된 위협요소에 대해서 STRIDE 분석 방법을 사용하였다. STRIDE 분석은 그림 5의 DFD 기반으로 그림 6과 같으며, 그 중 Sniffing을 통한 중요 데이터 탈취와 공격자가 생성한 임의의 Frame을 통한 Spoofing과 Firmware 변조 등의 위협을 최종적으로 식별하였다.

STRIDE	Explain
D	Jamming을 통해 Transfer Presentaion 프레임 전송을 방해하여 페어링 과정을 방해
S, T	노드가 전송하는 프레임의 NodeID 및 Command를 수정하여 다른 장비로 속인다.
S, T	AssignID를 수정한 Fake 프레임을 보내 강제 페어링
T	HomeID를 변조시킨 Fake AssignID 프레임을 보내 강제로 연결 해지
D	Find nodes in range 프레임을 지속적으로 보내어 명령 처리 대기 큐를 형성시켜 각 노드들에 대한 서비스 지연
T	모든 노드들의 Neighbor List 정보를 가진 컨트롤러의 Neighbor List 에 Fake 프레임을 보내 공격자가 변경한 정보 (노드 Neigbor List)로 변경
S	노드, 컨트롤러에 대한 인증이 Home ID 밖에 없기 때문에 공격자가 Nonce 를 정상적으로 요청
S, T, D	Jamming 또는 DoS 공격 후 공격자가 Fake Nonce Report 프레임을 각 노드에게 전달하여 공격자의 명령을 수행한다 .
I	Nonce Report 를 통해 전달되는 Nonce 값을 스니핑 하여 공격자가 정상적인 프레임을 생성할 때 사용
I	암호화에 사용하는 노드 장비 펌웨어 내 하드코딩 된 Passwc , Passwd 값을 공격자가 탈취 가능, 허브와 노드 간 패킷의 중요 정보 (HomdID, Command Class, Command, NodeID)를 Sniffing 하여 정상적인 암호문 및 인증 값을 생성
S	Firmware Updata Command Class를 이용해 각 노드를 악성코드에 감염
S	정상 프레임들을 Sniffing 한 후 Replay 패킷을 보내 노드 강제 제어

Fig. 6 Classification of identified threats using STRIDE model

V. 결론

본 논문에서는 Z-Wave 보안 프로토콜 Non-Secure, S0에 대해 취약점을 분석하고, 실제 환경을 구축하여 테스트하였다. S0 보안 프로토콜에서는 공정과정에서 HW에 셋팅되어 있는 $Passwd_c$, $Passwd_m$와 Sniffing을 통하여 원격 강제 제어가 가능한 상황이다. 또한, 본 논문에서 S0 프로토콜의 일부 프로세스에 대한 간략한 DFD를 나타내었으며, 차후 완성한 DFD를 바탕으로 STRIDE를 도출하여 Attack Tree를 구성할 것 이다. 추가적으로 프로토콜에 최적화된 새로운 STRIDE 모델까지 도출할 예정이다. 최종 향후 수행 목록으로 직접 구현한 Tool에 MITM, ARP Spoofing의 기술을 추가하여 단일 Tool 공격의 완성도를 높일 예정이다.

[Reference]

[1] Github. OpenZWave [Internet]. Available: https://github.com/OpenZWave.

[2] Badenhop, C. W., et al. "The Z-Wave routing protocol and its security implications." *Computers & Security*, vol. 68, pp. 112-129, Apr, 2017.

[3] Fouladi, B, and Sahand Ghanoun. "Security evaluation of the Z-Wave wireless protocol." *Black hat USA*, vol. 24, pp. 1-2, 2013.

[4] Digi-Key. Slicon Labs Dev Kit [Internet]. Available: http://bitly.kr/hdWS.

[5] ZWave Alliance. Z-Wave public specification [Internet]. Available: https://z-wavealliance.org/z-wave-public-specification/.